

1. Мы работаем с классификатором `classifier = linear_model.SGDClassifier()`. Мы хотим подобрать параметры модели с помощью поиска по сетке, а для этого нам хочется предварительно получить набор доступных параметров модели с их значениями. Какая из перечисленных команд позволяет это сделать?

1 point

- ☐ `classifier.params()`
- ☐ `classifier.params`
- ☐ `classifier.params_`
- ☒ `classifier.get_params()`

2. Какой из перечисленных параметров нельзя передать в `grid_search.GridSearchCV(...)` для подбора параметров модели по сетке?

1 point

- ☐ `scoring`
- ☒ `n_iter`
- ☐ `cv`

3. Мы работаем с объектом `grid_cv = grid_search.GridSearchCV(classifier, parameters_grid)`, реализующим функциональность поиска по сетке. После того, как мы провели поиск по сетке с помощью метода `fit`, нам нужно получить лучшую модель. Какая команда позволяет это сделать?

1 point

- ☐ `grid_cv.get_best_model()`
- ☐ `grid_cv.get_best_estimator()`
- ☐ `grid_cv.best_model`
- ☒ `grid_cv.best_estimator_`

4. Мы работаем с объектом `grid_cv = grid_search.GridSearchCV(classifier, parameters_grid)`, реализующим функциональность поиска по сетке. Здесь `classifier = linear_model.SGDClassifier()`. После того, как мы провели поиск по сетке с помощью метода `fit`, нам нужно получить набор лучших параметров модели `classifier`. Какая команда позволяет это сделать?

1 point

- ☐ `grid_cv.get_params()`
- ☐ `classifier.best_params_`
- ☐ `classifier.get_params()`

☒ `grid_cv.best_params_`

5. Какой класс позволяет построить цепочку преобразований для обработки данных? 1 point

- ☒ `sklearn.pipeline.Pipeline`
- ☐ `sklearn.pipeline.Chain`
- ☐ `sklearn.preprocessing.Pipeline`

6. Какую функциональность реализует объект `StandardScaler()` из модуля `sklearn.preprocessing`? 1 point

- ☐ Заполнение пропущенных значений в данных.
- ☒ Нормализация данных.
- ☐ Фильтрация данных согласно условию, передаваемому объекту `StandardScaler()` в качестве аргумента.